International Journal of Innovation and Scientific Research
ISSN 2351-8014 Vol. 22 No. 2 Apr. 2016, pp. 293-303

http://www.ijisr.issr-journals.org/

Implementation of High Speed & Low Power Approach by Designing Multi-Bit Flip-Flops

M. Manimaraboopathy[1], M. Anto Bennet[2], B. Sumitha[3], P. Nithyasri[3], and R. Pavithra[3]

[1]Assistant Professor, Department of Electronics and Communication Engineering, VELTECH, Chennai-600062, India

[2]Professor, Department of Electronics and Communication Engineering, VELTECH, Chennai-600062, India

[3]UG Student, Department of Electronics and Communication Engineering, VELTECH, Chennai-600062, India

ABSTRACT: Power has become a burning issue in modern VLSI design and integrated circuits; the power consumed by clocking gradually takes a dominant part. The proposed system provided a design to reduce the clock tree power by replacing some flip-flops with fewer multi-bit flip-flops, and also reduces the total power consumption. First, it perform a co-ordinate transformation to identify those flip flops that can be merged and also identify their legal regions in a library. Next step is to build a combination table to enumerate possible combinations of flip-flops provided by the library. The last step is to merge flip-flops in a hierarchical way. Besides power reduction, the objective of minimizing the total wire length is also considered. The time complexity of the proposed algorithm is less than the time complexity of the existing algorithm. According to the experimental results, the proposed algorithm significantly reduces the clock power by 27.9% and area reduced by 18.5%. The running time is very short. By using this method the low power consumed IC's can be manufactured using CMOS technologies.

KEYWORDS: Single & Double bit flip flops, Legal Placement Region, Flip flop Merging Power Report.

INTRODUCTION

In electronics, a flip-flop or latch is a circuit that has two stable states and can be used to store state information. A flip-flop is a bistable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in sequential logic. Flip-flops and latches are a fundamental building block of digital electronics systems used in computers, communications, and many other types of systems. Flip-flops and latches are used as data storage elements. Such data storage can be used for storage of state, and such a circuit is described as sequential logic. When used in a finite-state machine, the output and next state depend not only on its current input, but also on its current state (and hence, previous inputs). It can also be used for counting of pulses, and for synchronizing variably-timed input signals to some reference timing signal Flip-flops can be either simple (transparent or opaque) or clocked (synchronous or edge-triggered); the simple ones are commonly called latches. The word latch is mainly used for storage elements, while clocked devices are described as flip-flops. A latch is level-sensitive, whereas a flip-flop is edge-sensitive. That is, when a latch is enable it becomes transparent, while a flip flop's output only changes on a single type (positive going or negative going) of clock edge. Transparent latches are typically used as I/O ports or in asynchronous systems, or in synchronous two-phase systems.

LITERATURE SURVEY

Assem A. M. Bsoul and Steven J. E. Wilton (2010) had described a technique "An FPGA Architecture Supporting Dynamically Controlled Power Gating", which technique provides a modification to the fabric of an FPGA that enables dynamically-controlled power gating. It provides the total power consumption upto 23%.CorentinDupont et al(2012) had

described a technique "An Energy Aware Framework for Virtual Machine Placement in Cloud Federated Data Centres", which provides a flexible and energy-aware framework for the allocation of virtual machines in a data centre. This method provides 19% reduction in wire length. HoumanHomayouna,et al(2011) had described a technique "On leakage power optimization in clock tree networks for ASICs and general-purpose processors", which provides a post synthesis sleep transistor insertion (PSSTI), a heuristic clustering algorithm for sleep transistor insertion with the objective of total power minimization in a given clock tree. The clock tree leakage power is reduced by 19–32%. Jhen-Hong He, et al(2013) had described a technique "Clock Network Power Saving Using Multi-Bit Flip-Flops in Multiple Voltage Island Design", which provides an effective multi-bit flip-flop merging approach to deal with the clock network power minimization. it reduced the clock power up to 25%.

Mark Po-Hung Lin et al(2011) had introduced a technique "Post-Placement Power Optimization with Multi-Bit Flip-Flops", which describes a technique to reduce not only flip-flop power consumption but also clock tree and wire length. The power consumption obtained by 28%. Michael B. Henry(2011) had introduced a technique "Emerging Power-Gating Techniques for Low Power Digital Circuits", which provides an industry-standard technique, transistors are used to disconnect the power from idle portions of a chip. Present power-gating implementations suffer from limitations, which provides large amount of wasted energy.

Palden Lama et al(2012) had described a technique "Power-Aware Dynamic Placement and Migration in Virtualized GPU Environments", that controls the peak power consumption and improves the energy efficiency of server system. The result of this method is reduced power consumption by 23% and also reduction in power leakage.

Anto Bennet M et al(2015) had described a technique "High-Level Synthesis for Minimum-Area Low-Power Clock Gating", which describes an ILP (integer linear programming) formulation to consider both the clock tree and the clock control logic.. The overall power consumption is provided by this method is 32 Anto Bennet M et al(2015) et al(2012) had introduced a technique "Design Flow for Flip-Flop Grouping in Data-Driven Clock Gating", which describes a practical solution based on the toggling activity correlations of FFs. The data-driven clock gating is integrated into an Electronic Design Automation commercial Back end design flow, achieved power reduction of 15%–20%. Anto Bennet M et al(2014) had described a technique "Data-Width-Driven Power Gating of Integer Arithmetic Circuits", which method include a design that automatically implements coarse grain power-gated arithmetic circuits considering a narrow-width input data mode. This method provides 27% of power reduction. The demand for increased clock frequencies and logic availability (smaller area foot print) makes the problem even more important, leading among others to rapid elevation in power density. This literature survey spokes about the power, area consumption and reduction in wire lengths. Since power consumption is a critical challenge for implementing applications onto reconfigurable hardware. This proposed method provides the reduction in power, area consumption as well as the reduction in delay by merging many single bit flip-flops as a single multi bit flip-flop.

PROPOSED SYSTEM

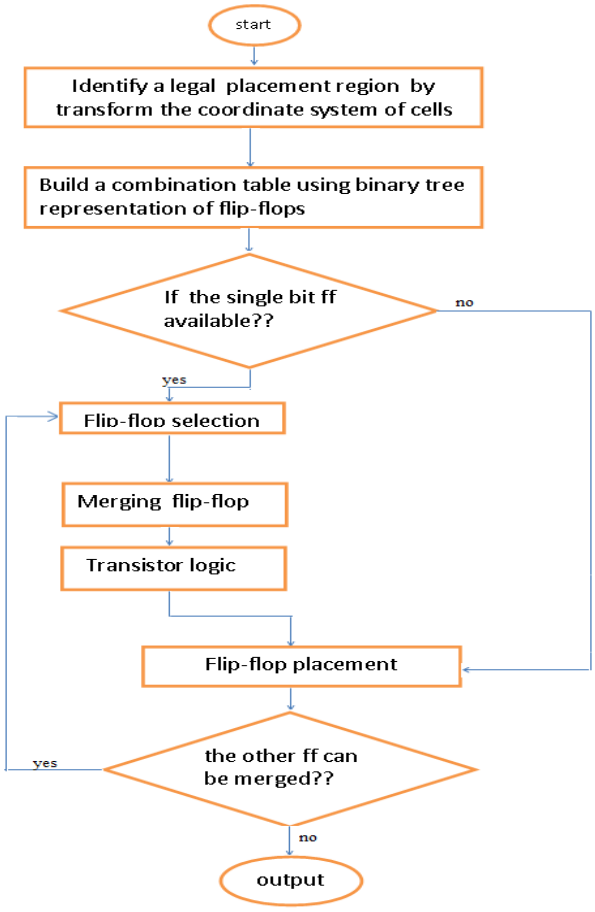


Fig 1 Flip-Flop Merging Flow Diagram

LEGAL PLACEMENT REGION

The shape of a feasible placement region associated with one pin denoted as pi connecting to a flip-flop denoted as fi. Since there may exist several pins connecting to fi,the legal placement region of fiare the overlapping area of several regions. Consider the two pins p1 and p2 connecting to a flip-flop f1, and the feasible placement regions for the two pins are enclosed by dotted lines, which are denoted by Rp(p1) and Rp(p2), respectively shown in fig 2. Thus, the legal placement region R(f1) for f1 is the overlapping part of these regions. However, it is not easy to identify and record feasible placement regions if their shapes are diamond.

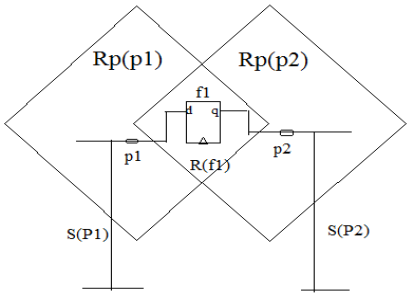


Fig 2. Legal Placement Region

The legal placement regions of flip-flop can be identified by using the following two methods Transformation of coordinate system & Determination of overlapped region.

TRANSFORMATION OF COORDINATE SYSTEM

The equations used to transform coordinate system are shown in eqn(1) and eqn(2). Suppose the location of a point in the original coordinate system is denoted by (x, y). After coordinate transformation, the new coordinate is denoted by (x', y'). In the original transformed equations, each value needs to be divided by the square root of 2, which would induce a longer computation time. Since it need to know the relative locations of flip-flops, such computation are ignored in this method. this method use x'andy' to denote the coordinates of transformed locations

$$x' = x + y \ /\sqrt{2} => x'' = \sqrt{2} * x' = x + y. \quad (1)$$

$$y' = -x + y \ /\sqrt{2} => y'' = \sqrt{2} * y' = -x + y. \quad (2)$$

DETERMINATION OF OVERLAPPED REGION

Then, it can find which flip-flops are mergeable according to whether their feasible regions overlap or not. Since the feasible placement region of each flip-flop can be easily identified after the coordinate transformation, simply use eqn(3) and eqn(4) to determine whether two flip-flops overlap or not.

$$DIS\_X(f1, f2) < 1/2 \ (W(f1) + W(f2)) \quad (3)$$

$$DIS\_Y(f1, f2) < 1/2(H(f1) + H(f2)) \quad (4)$$

Where,

W(f1) and H(f1) [W(f2) and H(f2)] denote the width and height of R(f1) [R(f2)], respectively, the function DIS_X(f1, f2) and (DIS_Y(f1, f2)) calculates the distance between centers of R(f1) and R(f2) in x& y directions.

BUILD A COMBINATION TABLE

If the system want to replace several flip-flops by a new flip-flop, system have to make sure that the new flip-flop provided by the library L when the feasible regions of these flip-flops overlap. In this paper, the method builds a combination table, which records all possible combinations of flip-flops to get feasible flip-flops before replacements. Thus, it can gradually replace flip-flops according to the order of the combinations of flip-flops in this table. Since only one combination of flip-flops needs to be considered in each time, the search time can be reduced greatly The pseudo code for building a combination table T. by using a binary tree to represent one combination for simplicity. Each node in the tree denotes one type of a flip-flop in L.For each node, the bit width of the corresponding flip-flop equals to the bit width summation of flip-flops denoted by its left and right child ,Let ni denote one combination in T, and b(ni) denote its bit width. In the beginning, initialize a combination ni for each kind of flip-flops in L. Then, in order to represent all combinations by using a binary tree, may add pseudo types, which denote those flip-flops that are not provided by the library . In order to use a binary tree to denote a type inL. If the combination is not included into any other combinations, it is deleted. First initialize two combinations n1 and n2 to represent these two types of flip-flops in the table T Next, the function Insert Pseudo Types performed to check whether the flip-flop types with bit widths between 1 and 4 exist or not. This is shown in algorithm 1.

ALGORITHM 1 BUILD COMBINATION TABLE

```
step1  : T = InitializationCombinationTable(L);
step2  : InsertPseudoType(L);
step3  : SortByBitNumber (L);
step4  : for each niin T do
step5  : InsertChildrens (ni, NULL, NULL);
step6  : index = 0;
step7  : whileindex!= size(T) do
step8  : Range_first= Rndex;
step9  : range_second= size(T);
step10: index = size(T);
```

step11: for each niin T
step12: for j = 1 to range_firstdo TypeVerify(ni, nj, T);
step13: for j = ito range_seconddo TypeVerify(ni, nj, T);
step14: T = DuplicateCombinationDelete(T);
step15: T = UnusedCombinationDelete(T);
InsertPseudoType(L):
step1 : for i= (bmin+1) to (bmax-1)
step2 : if (L does not contain a type whose bit width is equal to i)
step3 : insert a pseudo type typejwith bit width ito L;
InsertChildrens(n, n1, n2):
step1 : n.left_child← n1;
step2 : n.right_child← n2;
TypeVerify(n1, n2, T):
step1 : bsum= b(n1) + b(n2);
step2 : if (L contains a type whose bit width is bsum)
step3 : insert a new combination n whose bit width bsumto T;

FINAL TABLE

By combining two 1-bit flip-flops in the first combination, a new combination n3 can be obtained. Similarly, a new combination n4 (n5) can be easily obtain by combining n1 and n3(two n3's) Finally, n6 is obtained by combining n1 and n4. To speed up this program, n6 is deleted from T rather than n5 because its height is larger. After this procedure, n4 becomes an unused combination since the root of binary tree of n4 corresponds to the pseudo type, type3, in Land it is only included in n6. After deleting n6, n4 is also need to be deleted. The last combination table This shown in table 1. In order to enumerate all possible combinations in the combination table, all the flip-flops whose bit widths range between bmax and bmin and do not exist in Lshould be inserted into L.

Table 1 Combination Table

Combination Table *T*			
n_1	n_2	n_3	n_4
1-bit	4-bit	2-bit	4-bit
		n_1 + n_1	n_3 + n_3

There exist several choices if want to build a binary tree corresponding to a type type j. However, the complete binary tree has the smallest height. Thus, for building a binary tree of a certain combination ni whose type is type j, only the flip-flops whose bit widths are (b(type j)/2) and (b(type j)b(type j)/2) should exist in **L**. which is shown in algorithm 2.

ALGORITHM 2 INSERT PSEUDO TYPES

InsertPseudoType(L):
step1 : for each typejin L do
step2 : Pseudo Type Verify Insertion(typej, L) ;
Pseudo Type Verify Insertion(typej, L):
step1 : if (mod (b(typej) /2) == 0)
step2 : b1 = [b(typej)/2], b2 = [b(typej)/2];
step3 : else
step4 : b1 = [b(typej)/2], b2 = b(typej) – [b(typej)/];

step5 : for i= 1 to 2
step6 : if ((bi >bmin) &&
(L does not contain a type whose bit width is equal to bi))
step7 : insert a pseudo type typejwith bit width bi to L;
step 8 :Pseudo Type Verify Insertion(typej, L);

Insertion recursively checks the existence of flip-flops whose bit widths around b(type j)/2and add them into Lif they do not exist. The function Pseudo Type Verify Insertion, it divides the bit width b(type j) into two parts b(type j)/2 and b(type j)/2, (b(typej)/2 and b(type j), b(type j)/2) if b(type j) is an even (odd) number), and it would insert a pseudo type type j into L if the type is not provided by Land its bit width is larger than the minimum bit width (denoted by bmin) of flip-flops in L (see Lines 5–8 in Pseudo Type Verify Insertion). The same procedure repeats in the new created type. Note that this method works only when the 1-bit type exists in L.For example, assume a library Lonly provides two kinds of flip-flops whose bit widths are 1 and 7. In the new procedure, it first adds two pseudo types of flip-flops whose bit widths are 3 and 4, respectively, for the flip-flop with 7-bit.

MERGE FLIP-FLOPS

Use of the combination table is to combine flip-flops in this subsection. To reduce the complexity, first divide the whole placement region into several sub regions, and use the combination table to replace flip-flops in each sub region. Then, several sub regions are combined into a larger sub region and the flip-flops are replaced again so that those flip-flops in the neighboring sub regions can be replaced further. Finally, those flip-flops with pseudo types are deleted in the last stage because they are not provided by the supported library.Region Partition: To speed up our problem, the whole chip into several sub regions. Replacement of Flip-flops in Each Sub region: Before illustrating this procedure to merge flip-flops, first give an equation to measure the quality if two flip-flops are going to be replaced by a new flip-flop as follows:

$$\text{cost} = \text{routing length} - \alpha \times \text{available area}.$$

Where routing length denotes the total routing length between the new flip-flop and the pins connected to it, and available area represents the available area in the feasible region for placing the new flip-flop.α is a weighting factor The cost function includes the term routing length to replacement that induces shorter wire length. Besides, if the region has larger available space to place a new flip-flop, it implies that it has higher opportunities to combine with other flip-flops in the future and more power reduction. Once the flip-flops cannot be merged to a higher-bit type ignore the available area in the cost function, and hence α is set to 0. Bottom-Up Flow of Sub region Combinations: there may exist some flip-flops in the boundary of each sub region that cannot be replaced by any flip-flop in its sub region. However, these flip-flops may be merged with other flip-flops in neigh boring sub regions. Hence, to reduce power consumption furthermore, it can combine several sub regions to obtain a larger sub region and perform the replacement again in the new sub region again. The procedure repeats until it cannot achieve any replacement in the new sub region. Suppose divide a chip into 16 sub regions in the beginning. After the replacement of flip-flops is finished in each sub region, four sub regions are combined to get a larger Suppose some flip-flops in new sub regions still can be replaced by new flip-flops in other new sub regions, would combine four sub regions to get a larger one and perform their placement in the new sub region again. As the procedure repeats in a higher level, the number of merge able flip-flops gets fewer. However, it would spend much time to get little improvement for power saving. To consider this issue, there exists a trade-off between power saving and time consuming in this program.De-Replace and Replace: Since the pseudo type is an intermediate type, which is used to enumerate all possible combinations in the combination table T, it has to remove the flip-flops belonging to pseudo types. Thus, after the above procedures have been applied, it would perform de-replacement and replacement functions if there exists any flop-flops belonging to a pseudo type. For example, if there still exists a flip-flop, fi, belonging to n3 after replacements it have to de-replace fi into two flip-flops originally belongs to n1. After de-replacing, it do the replacements of flip-flops according to T without consideration of the combinations whose corresponding type is pseudo in L.

POWER EFFICIENCY

The modification of the multi-bit flip-flop is to implement in the transistor logic and here this technique used D_Flipflop basis of transistor operation and it processed depends on both clock and data inputs. This method use the both N-mos and P-mos transistor logic acts the D_Flipflop and whiles during the operation the X-node is the data transferring the next stage and holding the data by using the Delay inverters.

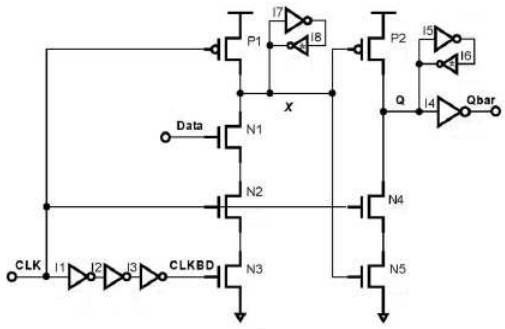


Fig 3 D-Flip-Flop Operation Through Transistor

Fig 3 shows the D-Flip-flop operation through transistor and in/out concept this method using multiple bits so the same concept can be apply for 2-bit operation it was produced the corresponding result and by using the merging method it can able to reduces the power and it was consumed less power shown in table 3.

RESULT AND DISCUSSION

As shown in table 2, the proposed system results of Power and Area are better when compare to the existing method results. The total power consumption for existing method is 68mW and it is reduced as 49mW in the proposed method. The area is reduced from 27(µm)² to 22(um)².

Table 2 Proposed Method Power, Area Comparison

Parameter	Existing method	Proposed method
Power	68mW	49mW
Area	27(µm)²	22(µm)²

Table 3 Single Bit FF Input/output

CLOCK1	CLOCK2	D1	D2	Q1	Q2
0	0	0	0	0	1

SINGLE BIT FLIP-FLOP O/P WAVE FORM

It is a wave form of two single bit flip-flops with two individual clocks shown in fig 4.

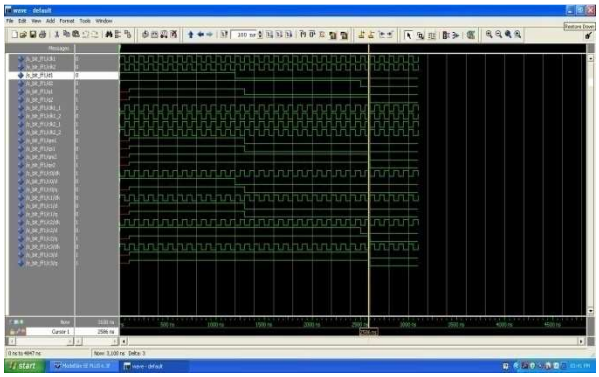


Fig 4 Single Bit Flip-Flop Output Waveform.

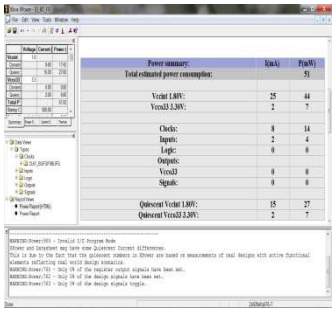


Fig 5 Single Bit Flip-Flop Power Report

SINGLE BIT FLIP-FLOP POWER REPORT

It [Fig 5] shows the total power consumption of two flip-flops at different clock pulses

DOUBLE BIT FLIP-FLOP WAVE FORM

It shows the wave form of double bit flip-flops operated by using a single clock pulse shown in fig 6.

Table 3 MBFF Input/output

CLOCK1	D1	D2	Q1	Q2
0	0	1	0	1

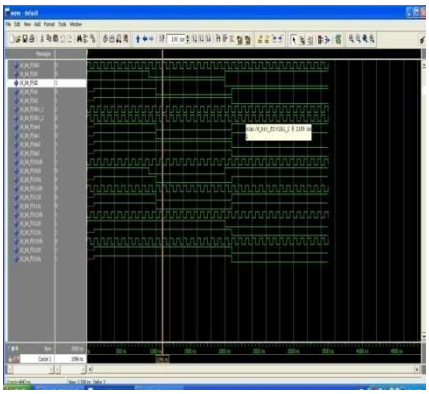


Fig 6 Double Bit Flip-Flop Output Waveform.

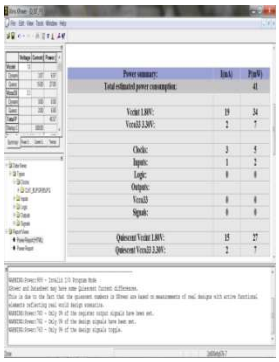


Fig 7Double Bit Flip-Flop Power Report

DOUBLE BIT FLIP-FLOP POWER REPORT

It shows the total power consumption of two flip-flops at same clock pulse shown in fig 7.

COMBINATIONAL TABLE OUTPUT WAVE FORM

It shows the wave form of combinational table operated by using a single clock pulse shown in fig 8.

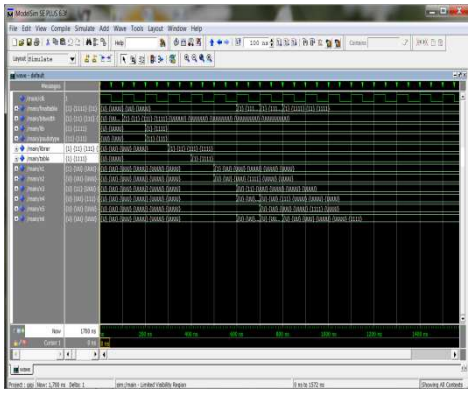
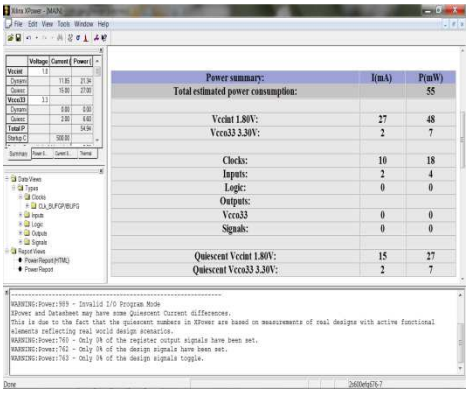


Fig 8.CombinationalTable Output Waveform.



Power summary:	I(mA)	P(mW)
Total estimated power consumption:		55
Vccint 1.80V:	27	48
Vcco33 3.30V:	2	7
Clocks:	10	18
Inputs:	2	4
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

Fig 9.Combinational Table Power Report.

COMBINATION TABLE POWER REPORT

It shows the total power consumption of combinational table at one clock pulse.

FLIP-FLOP MERGING WAVEFORM

It shows the wave form of merged flip-flops operated by using a single clock pulse shown in fig 10.

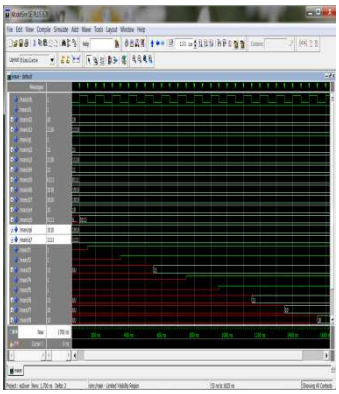
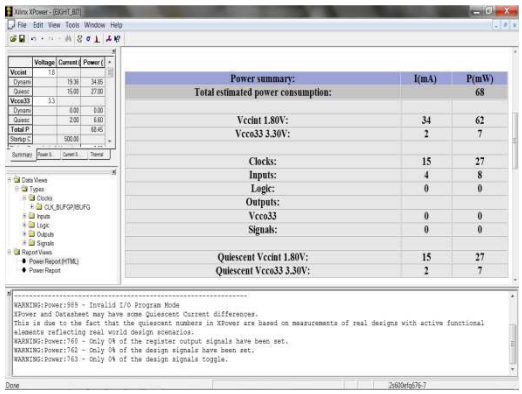


Fig 10 Flip-Flop Merging Output Waveform.



Power summary:	I(mA)	P(mW)
Total estimated power consumption:		68
Vccint 1.80V:	34	62
Vcco33 3.30V:	2	7
Clocks:	15	27
Inputs:	4	8
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

Fig 11 Flip-FlopMerging Power Report

FLIP-FLOP MERGING POWER REPORT

It shows the total power consumption of merging at one clock pulse shown in fig 11.

FLIP-FLOP MERGING RTL SCHEMATIC VIEW

It is a RTL SCHEMATIC VIEW of the 8-bit merging flip-flops. It shows the internal connections of the flip-flops shown in fig 12

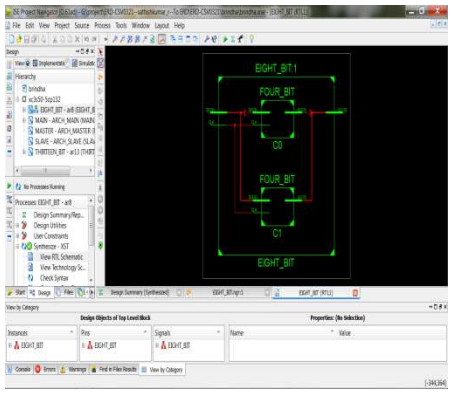


Fig 12Flip-Flop Merging RTL Schematic View Output.

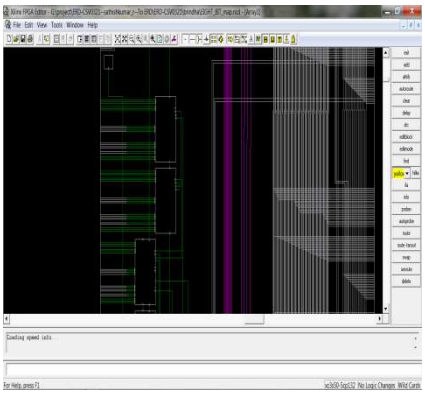


Fig 13Flip-Flop Merging Technological View Output.

FLIP-FLOP MERGING TECHNOLOGICAL VIEW

It is a [13] diagram shows that the internal circuit connections of flip-flop merging method.

FLIP-FLOP MERGING AREA MAPPING

This [14] diagram shows the area mapping of the flip-flops in the circuit by using PLAN AHEAD software.

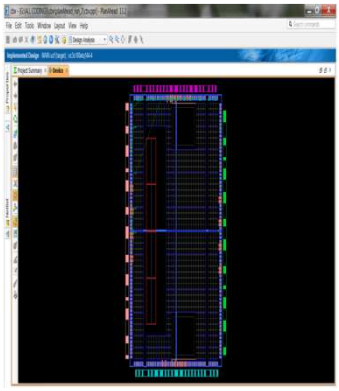


Fig 14 Flip-Flop Merging Area Mapping Output.

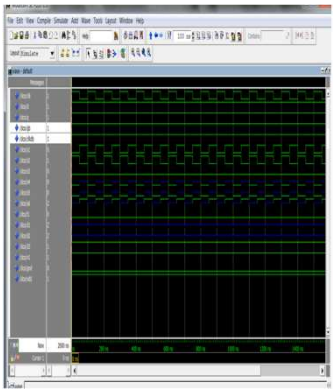


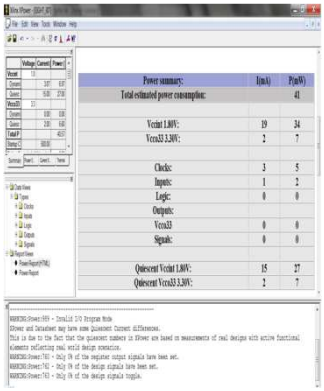
Fig 15 Transistor Logic Output Waveform.

TRANSISTOR LOGIC WAVEFORM

It [15] shows the wave form of transistor logic operated by using a single clock pulse.

TRANSISTOR LOGIC POWER REPORT

It [16] shows the total power consumption of transistor logic at one clock pulse.



Power summary:	I(mA)	P(mW)
Total estimated power consumption:		41
Vccint 1.80V:	19	34
Vcco33 3.30V:	2	7
Clocks:	3	5
Inputs:	1	2
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

Fig 16Power Report ForTransistor Logic.

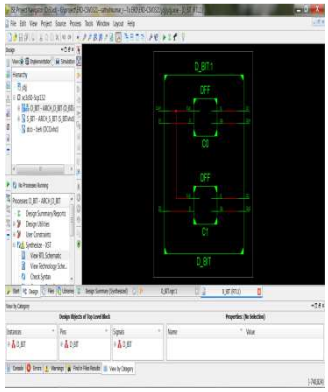


Fig 17 RTL Schematic View Output For Transistor Logic

TRANSISTOR LOGIC RTL SCHEMATIC VIEW

It is a RTL SCHEMATIC VIEW of the merged flip-flop proposed work. It [Fig 17] shows the internal connections of the flip-flops.

TRANSISTOR LOGIC TECHNOLOGICAL VIEW

It is a diagram [18] shows that the internal circuit connections of the transistor logic design.

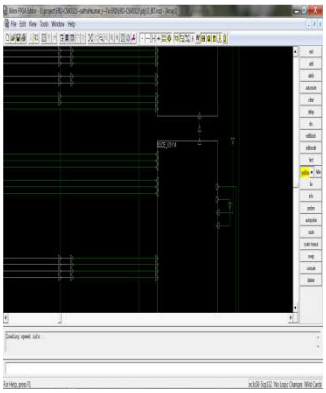


Fig 18 Technological View Output of Transistor Logic

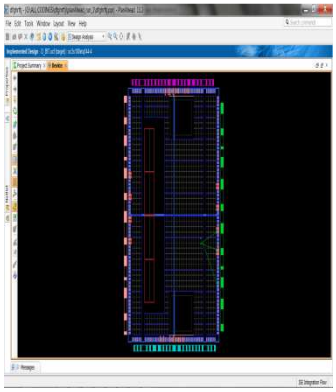


Fig 19 Transistor Logic Area Mapping Output.

TRANSISTOR LOGIC AREA MAPPING

This [19] diagram shows the area mapping of the flip-flops in the circuit by using PLAN AHEAD software.

CONCLUSION

In this proposed system numbers of single bit flip-flops are merged as a multi bit flip-flop for the purpose of power reduction.The procedure of flip-flop replacements is depending on the combination table, which records the relationships among the flip-flop types. The concept of pseudo type is introduced to help to enumerate all possible combinations in the combination table. By the guidelines of replacements from the combination table, the impossible combination of flip-flops is not being considered that decreases execution time. The proposed results achieved power reduction upto 27.9%, area reduction upto 18.5%. The proposed system provides the flip-flop merging concept only by using D type flip-flops. The future work of this project contains merging the other types of flip-flop like T flip-flop, SR flip-flop, JK flip-flop. This merging concept will applied in the transistor logic to obtain high power reduction.

REFERENCES

[1] Assem A. M. Bsoul, Steven J. E. Wilton (2010), "An FPGA Architecture Supporting Dynamically Controlled Power Gating" in Proc. ACM/IEEE Des. Autom. Conf., pp. 176–181.

[2] Corentin Dupont, Giovanni Giuliani, Fabien Hermenier (2012)," An Energy Aware Framework for Virtual Machine Placement in Cloud Federated Data Centres," IEEE Trans. Computer[20] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation,"IEEE Trans. Comput., vol. C-35, pp. 677–691.

[3] Houman Homayouna, Shahin Golshanb (2011), "On leakage power optimization in clock tree networks for ASICs and general-purpose processors", IEEE Trans. Circuits Syst.I,vol. 47, no. 3, pp. 415–420.

[4] Jhen-Hong He, Li-Wei Huang, Jui-Hung Hung, Yu-Cheng Lin, Guo-syuan Liou, Tsai-Ming Hsieh (2013), "Clock Network Power Saving Using Multi-Bit Flip-Flops in Multiple Voltage Island Design," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Des., pp. 30–37.

[5] Mark Po-Hung Lin, Chih-Cheng Hsu, Yao-Tsung Chang (2011), "Post-Placement Power Optimization with Multi-Bit Flip-Flops," in Proc. ACM/IEEEDes. Autom. Conf., pp. 159–164.

[6] Michael B. Henry (2011), "Emerging Power-Gating Techniques for Low Power Digital Circuits," in Proc. ACM/IEEE Des. Autom. Conf., pp. 795–800.

[7] Palden Lama, Yan Li, Ashwin M. Aji, Pavan Balaji (2012), "Power-Aware Dynamic Placement and Migration in Virtualized GPU Environments," IEEE Trans. Comput., vol. C-35, pp. 677–691.

[8] Dr. Anto Bennet, M, Sankar Babu G, Suresh R, Mohammed Sulaiman S, Sheriff M, Janakiraman G, Natarajan S, "Design & Testing of Tcam Faults Using T_H Algorithm", Middle-East Journal of Scientific Research 23(08): 1921-1929, August 2015 .

[9] Dr. Anto Bennet, M "Power Optimization Techniques for sequential elements using pulse triggered flipflops", International Journal of Computer & Modern Technology , Issue 01 ,Volume01 ,pp 29-40, June 2015.

[10] Dr. Anto Bennet, M, Manimaraboopathy M,P. Maragathavalli P, Dinesh Kumar T R, "Low Complexity Multiplier For Gf(2m) Based All One Polynomial", Middle-East Journal of Scientific Research 21 (11): 2064-2071, October 2014.